

Desafio Front-End

Desafio Técnico — Desenvolvedor(a) Front End (React)

1. Tema

Gestão de Cursos Online Colaborativa — **CourseSphere**.

2. Objetivo

Desenvolver uma aplicação web em **React com hooks** com foco em:

- Autenticação de usuários.
 - CRUD com controle de permissões.
 - Busca com filtros e paginação.
 - Consumo de API local e API externa pública.
 - Organização clara do projeto e qualidade de código.
 - Interface responsiva e boa experiência de usuário (UX).
-

3. Critérios de Avaliação

3.1 Prioridade Alta

- Organização e estrutura do projeto.
- Uso adequado de hooks.

3.2 Prioridade Média

- Qualidade visual (layout, responsividade, UX).
- Permissões e regras de acesso implementadas corretamente.
- Validações de formulários.
- Componentização.
- Consumo correto das APIs.
- Feedbacks de interação (carregamento, erros, sucesso).

3.3 Prioridade Baixa

- Completude (todas as funcionalidades descritas implementadas).
-

4. Funcionalidades

4.1 Autenticação

- Tela de login com e-mail e senha.
- Usuários não autenticados não podem acessar páginas protegidas (cursos, aulas, etc.).

4.2 Entidades e Validações

4.2.1 User

Atributo	Regra
name	Obrigatório.
email	Obrigatório; formato válido; único.
password	Obrigatório; mínimo 6 caracteres.

Regras gerais

- Apenas usuários autenticados acessam a plataforma.

4.2.2 Course

Atributo	Regra
name	Obrigatório; mínimo 3 caracteres.
description	Opcional; até 500 caracteres.
start_date	Obrigatório; data válida.
end_date	Obrigatório; posterior a <code>start_date</code> .
creator_id	ID do instrutor criador.
instructors	Lista de instrutores colaboradores.

Regras

- Apenas o **criador** pode **editar/excluir** o curso.
- Apenas o **criador** pode **gerenciar a lista de instrutores**.

4.2.3 Lesson

Atributo	Regra
----------	-------

title	Obrigatório; mínimo 3 caracteres.
status	Obrigatório; valores válidos: <code>draft</code> , <code>published</code> , <code>archived</code> .
publish_date	Obrigatório; deve ser data futura.
video_url	Obrigatório; URL de vídeo válida.
course_id	ID do curso ao qual pertence.
creator_id	ID do instrutor criador da aula.

Regras

- Qualquer **instrutor** do curso pode **criar** aulas.
- Apenas o **criador da aula** ou o **criador do curso** pode **editar/excluir** a aula.

4.3 Busca e Filtros (Listagem de Aulas)

- Campo de busca por `title`.
- Filtro por `status` (`draft`, `published`, `archived`).
- Filtro por `COURSE`.
- Todos os filtros devem funcionar em conjunto.
- Resultados **paginados**.

4.4 Consumo de APIs

- **API local** (ex.: JSON Server) para `users`, `courses`, `lessons`.
- **API externa pública** (ex.: <https://randomuser.me>) para sugerir novos instrutores.

5. Telas e Funcionalidades Esperadas

5.1 Tela de Login

- Formulário com validações de e-mail e senha.
- Redireciona para o **Dashboard** após login.
- Bloqueia rotas protegidas para não autenticados.

5.2 Dashboard (Tela Inicial)

- Lista de cursos em que o usuário é **criador** ou **instrutor colaborador**.

- Cards com `name`, `description`, `start_date`, `end_date`.
- Ação para **criar novo curso**.

5.3 Detalhes do Curso

Exibe:

- `name`, `description`, `start_date`, `end_date`.
- Lista de **instrutores**.
- Lista de **aulas** com:
 - Miniatura do vídeo (derivada de `video_url`).
 - **Busca, filtros e paginação**.

Ações:

- **Criador**: editar/excluir curso; gerenciar instrutores; editar/excluir **qualquer** aula.
- **Instrutores**: criar aulas; editar/excluir **as próprias** aulas.

5.4 Cadastro/Edição de Curso

- Formulário com `name`, `description`, `start_date`, `end_date`.
- Validações: campos obrigatórios; `end_date` > `start_date`.
- **Acesso restrito** ao criador para editar/excluir.

5.5 Cadastro/Edição de Aula

- Formulário com `title`, `status`, `publish_date`, `video_url`.
- Validações obrigatórias conforme seção 4.2.3.
- Permissões: criador da aula ou criador do curso; qualquer instrutor pode criar.

5.6 Gerenciamento de Instrutores

- O **criador do curso** pode:
 - Visualizar e **remover** instrutores.
 - **Adicionar** novos instrutores via API externa.
 - Exibir feedback visual de sucesso/erro nas ações.

5.7 Tela de Acesso Negado ou Erro (Opcional — Diferencial)

- Informa o usuário sobre permissões insuficientes ou erros inesperados.
-

6. Diferenciais

- Estrutura seguindo **Atomic Design**.
 - Commits com **mensagens semânticas**.
 - Layout **responsivo** e agradável.
 - **Feedbacks visuais**: loaders; mensagens de sucesso/erro.
 - Uso ampliado de **APIs externas**.
 - **Deploy** funcional (Netlify, Vercel, etc.).
-

7. Stack

- **React com hooks** (obrigatório).
 - Demais bibliotecas e backend: livres.
-

8. Entrega

- Repositório público no GitHub.
- **README** com:
 - Instruções de instalação e execução.
 - (Opcional) link de deploy online.